

# Keyframe-Based User Interfaces for Digital Video



**Three visual interfaces that use keyframes—still images automatically pulled from video—to provide access points for efficient navigation of recorded content can help identify potentially useful or relevant video segments.**

Andreas  
Girgensohn

John  
Boreczky

Lynn Wilcox  
FX Palo Alto  
Laboratory

As video use expands from entertainment to other domains—such as business, education, and personal applications—these new uses demand new forms of access. For entertainment, the typical user will happily view video linearly, making current access methods—such as fast forward and rewind—sufficient. However, in other applications, users may need to skip around to seek specific video segments. For example, a user may want to find a particular meeting in a database containing a number of recorded meetings. Or, in a recorded seminar, the user may want to view a segment where participants discussed a certain topic. In a personal video collection, the user may simply seek a particular piece of footage.

To meet these diverse needs, we describe three visual interfaces<sup>1-3</sup> that help people identify potentially useful or relevant video segments. In such interfaces, *keyframes*—still images automatically extracted from video footage—can distinguish videos, summarize them, and provide access points. Well-chosen keyframes help users select videos and enhance a listing's visual appeal. Our goal is to identify keyframes that describe an entire video and provide access to its relevant parts.

Keyframe selection can vary depending on the application's requirements. For example, a visual summary of a video-captured meeting only needs a few keyframes that show the highlights, whereas a video editing system needs a keyframe for every clip. Browsing interfaces require a somewhat even distribution of keyframes over the length of the video.

Selecting good keyframes is difficult. Most systems select at least one keyframe for each *shot*. A shot is captured video from a single camera running from camera on to camera off. Using one keyframe per shot means that representing a one-hour video usually requires hundreds of keyframes. In contrast, our approach for video indexing and summarization selects fewer keyframes that represent the entire video and index the interesting parts. The user can select the number of keyframes or the application can select the optimal number of keyframes based on display size, but a one-hour video typically will have between 10 and 40 keyframes.

We use several techniques to present the automatically selected keyframes. A video directory listing shows one keyframe for each video and provides a slider that lets the user change the keyframes dynamically. The visual summary of a single video presents images in a compact, visually pleasing display. To deal with the large number of keyframes that represent clips in a video editing system, we group keyframes into *piles* based on their visual similarity. In all three interfaces, the user can start playback at any specified point by clicking on a particular frame.

Our employees have been using the keyframe slider and video summarization component on a regular basis as part of a larger video database system<sup>3</sup> for more than two years. The database contains a large collection of videotaped meetings and seminars, as well as videos from other sources, for example, video demonstrations. The clip-browsing interface forms

part of a home video-editing system.<sup>2</sup> The “User Studies” sidebar describes the studies we conducted to compare different design alternatives and to examine the system’s usability. We have made several improvements to the system based on the results of these studies.

### DIRECTORY LISTINGS

When users access videotapes of meetings and seminars, they often face the problem of determining which of several videos contains a specific event. Even after they identify the correct video, accessing the section they need can be difficult. To address this problem, we implemented a Web-based browser that presents directory listings of videos<sup>3</sup> organized by content category, such as staff meetings or video demonstrations. This listing provides a skimming

interface that makes it easy to compare videos and helps users find a good starting point for video playback.

The skimming interface has a keyframe window for each video. The user changes the keyframe in the window by moving the mouse along a timeline. Blue triangles mark the positions of the keyframes along the timeline. As the mouse pointer—shown as a hand cursor in Figure 1—moves over the timeline, a slider thumb shows the position on the timeline, the keyframe closest to the mouse position displays, and the triangle for that keyframe turns red.

This method shows only a single keyframe at a time, preserving screen space while making other frames accessible through simple mouse motion. The interface supports very quick skimming to provide an overall impression of the video’s content. Clicking

## User Studies

We conducted user studies for each of three user interfaces: keyframe slider, pictorial summary, and video clip browser. The first two studies compared different design alternatives. In both studies, the participants expressed a preference for our user interface over the alternatives not using our techniques. The last study examined the general usability of the system and found that the participants could use it without problems.

### Keyframe slider

For the keyframe slider interface, we conducted a small study to observe user behavior during a set of typical browsing tasks. One group of participants used a simple Web-based directory listing with a single keyframe for each video. The other group used our keyframe slider interface. We created information retrieval tasks representative of the typical activities of our users. Participants used a wide variety of strategies to find the required information in the video documents. This led to a large variation in task completion times so that there was no significant difference in performance between the two groups. After the participants completed the tasks, we asked them questions about the usefulness of the various features. Both feedback and observed behavior indicate the usefulness of multiple keyframes.

### Pictorial summary

We designed different styles to explore two features of our video summaries in the manga comic book style: image selection by importance score and variable-size image layout. The first style (control) used neither feature: The display consisted of fixed-size images sampled at regular time intervals. The second style (selected) used our importance score for image selection but retained fixed-size images. The third style (manga) used both features. Participants answered questions by finding the relevant video segments. There was no significant difference in the task completion time across the three variants. However, the participants completed the tasks in a small fraction of the

time needed for watching the entire videos (8.6 percent).

In the second part of the study, we asked the participants for their judgments regarding the suitability of combinations of our techniques for summaries and navigation. We also asked them to report their overall preferences regarding the visual appeal of the summaries. They looked at pairs of different summaries for the same video and answered the following questions for each pair:

- Which one more effectively summarizes the video?
- Which one provides better entry points into the video?
- Which one is more visually pleasing?
- Which one do you like better?

The participants judged the manga style to be superior to other conditions for all questions. Because the summaries using our importance score for selecting fixed-size images were not considered superior to the summaries of the control style, the variable-size layout seems to be more important than the keyframe selection.

### Video clip browser

The third study examined our video editing system and its use of clips and piles. The participants edited footage they had shot. In the editing sessions, all participants worked consistently and created output films. Participants said they liked using our editor, particularly the ability to play clips and to drag and drop them easily. Participants commented that they liked the process and found it a quick and easy way to get interesting parts out of the raw footage and to arrange them for viewing. The participants indicated no problems with the overall notion of piles. Comments and observations indicated that having multiple views to get different perspectives on the video footage was beneficial. Participants could easily flip through images, often getting a sense of the clips without watching the video itself.

anywhere on the timeline opens the video and starts playback at the corresponding time. Clicking on the keyframe also starts playback exactly at that keyframe. Using multiple keyframes in this way gives users an idea of the video's context and temporal structure.

Because keyframes attach to the timeline, we want them spread out over time. To support skimming through a video, a semiuniform distribution of keyframes that keeps them from grouping too closely together works best.

We address these issues by selecting representative frames from a collection of similar frames that might be distributed throughout the video. In the slider interface, zooming in or out modifies the number of keyframes. Zooming modifies the width of the timeline onscreen. The average spacing between keyframes on the screen remains constant so that increasing the width of the timeline also increases the number of attached keyframes, as Figure 1 shows.

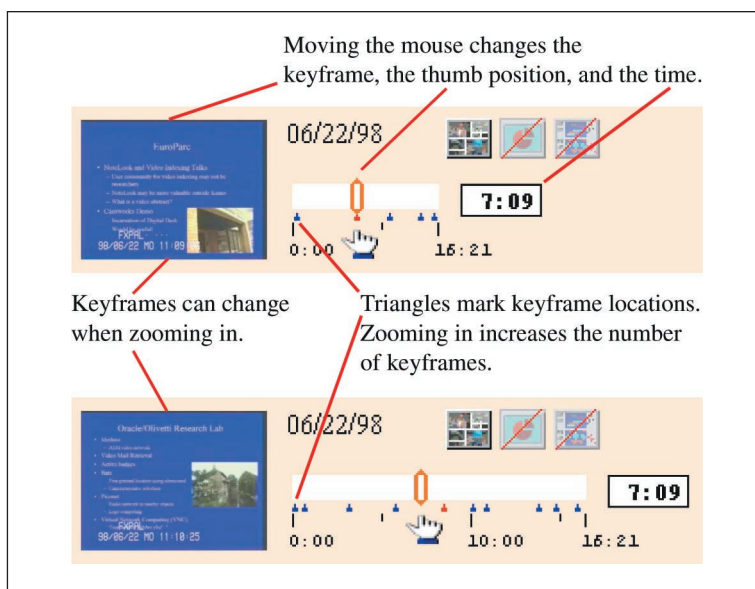
Other systems that use keyframes to support video browsing either provide limited control over the number of keyframes or do not find truly representative frames. Most of these systems break videos into shots and select the frame closest to the center as each shot's keyframe.<sup>4</sup> Yueting Zhuang and colleagues<sup>5</sup> used an approach that clusters keyframes to represent shots that have more interesting visual content, but it still extracts at least one keyframe per shot.

Instead of segmenting the video into shots, our technique clusters all of the video's frames. When selecting a set of keyframes that differ from one another but still represent the whole video, taking one frame from each cluster meets that goal. No shot detection is necessary in this approach, and the number of keyframes can be changed easily.

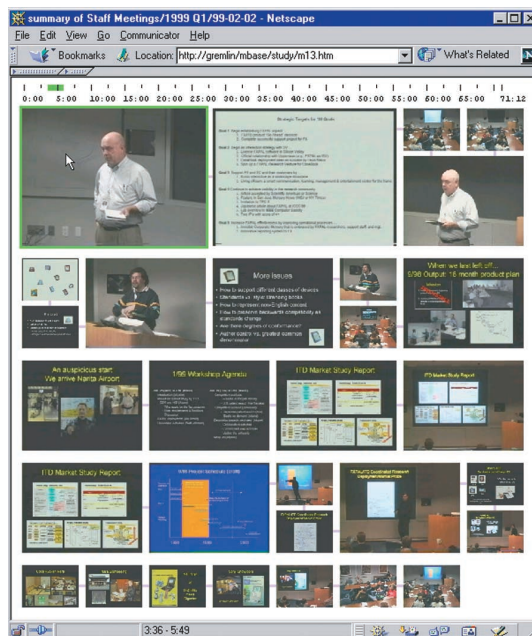
## VIDEO SUMMARIES

Because video is a linear medium, it's difficult to get an overview of a video's content without watching it at either normal or high speed. For videos that contain enough visible action, a well-designed visual summary displayed on a single page can address this problem. Typical approaches summarize a video by segmenting it into shots demarcated by camera changes. Then, those approaches can represent the entire video as a collection of keyframes, with one keyframe for each shot.

Although this technique reduces the amount of information a user must sift through to find the desired segment, it may still generate too much data. Most approaches use a basically linear frame presentation, although some occasionally use other structures.<sup>6</sup> These systems produce summaries as an aid to navigation, not as an attempt to distill the essential content. Further, the relative similarity of keyframes,



**Figure 1. Time slider for displaying keyframes.** As the hand cursor moves over the timeline, a slider thumb shows the position on the timeline, the keyframe closest to the cursor displays, and the triangle for that keyframe turns red.



**Figure 2. Pictorial summary of a video recording.** The system discards redundant information, such as repeating or alternating shots, creating an overview that also serves as a navigation tool.

coupled with a large regular display, can make it harder to spot the desired keyframe.

In contrast, as Figure 2 shows, our system abstracts video by selectively discarding or deemphasizing redundant information, such as repeated or alternating shots. This approach presents a concise summary that provides an overview of the video and serves as a navigation tool.

Michael Christel and colleagues<sup>7</sup> described a system that automatically produces video highlights by selecting short video segments. For each shot, the system selects a keyframe that emphasizes moving objects, faces, and text. Their algorithm then selects shot keyframes in rank order and adds a short segment of video surrounding the selected keyframe to the video highlight until the highlight reaches the desired length. This approach favors interesting keyframes, but it loses the video's structure.

### Selecting keyframes for the summary

To select keyframes, we calculate an importance score for each segment. To produce a good summary requires discarding or deemphasizing many segments. Thresholding the importance score prunes less important shots, leaving a concise and visually varied summary. We calculate the importance score for each segment based on its rarity and duration—longer shots are likely to be more important. The emphasis on longer shots also avoids including video artifacts such as synchronization problems after camera switches. At the same time, even if they are long, repeated shots—such as wide-angle shots of a conference room—receive lower scores because they do not add much to the summary. To generate a pictorial summary, we select segments with an importance score higher than a threshold. We extract the frame nearest each selected segment's center as a representative keyframe.

The elements in our summaries vary both in content and keyframe size. To draw attention to the

video's most important segments, we size frames according to the importance measure of their originating segments, and larger keyframes represent higher-importance segments.

We pack the different-sized keyframes into rows and resize them to best fit the available space while maintaining their time order. The result is a compact, visually pleasing summary reminiscent of the sequential images that define the comic book and Japanese manga art forms.

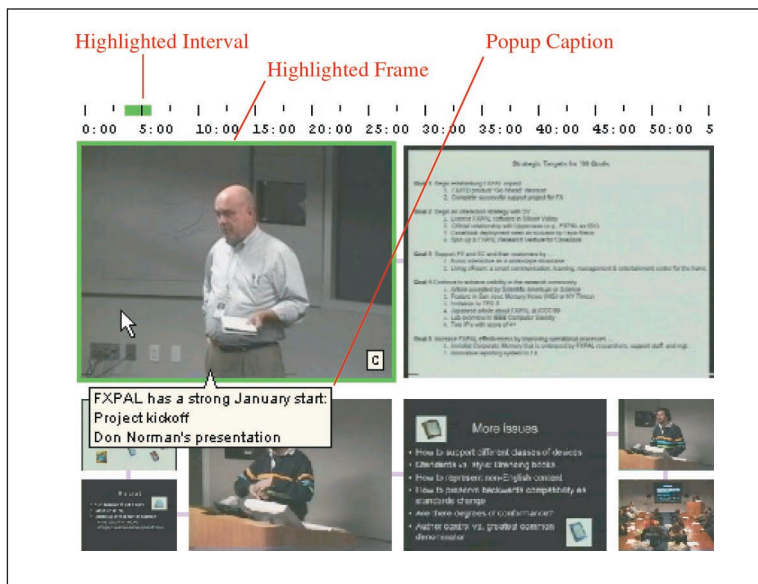
### Exploring videos

A summary can help a user identify the desired video simply by showing relevant images, but sometimes this does not provide enough information. Even after finding an appropriate video, locating specific passages within it simply by inspection can be difficult.

Interactive browsing of video summaries can shorten the time required to find the desired segment. To facilitate this process, we have implemented a Web-based interactive version of the pictorial summary that uses either keyframes or the video's timeline to browse a video. The two views always synchronize. The timeline shows the duration of the segment that corresponds with the frame under the cursor, as Figure 3 shows. Similarly, when the cursor hovers over the timeline, the corresponding keyframe is highlighted. This display lets users explore a video's temporal properties. At a glance, users can see both the visual representation of an important segment and its corresponding time interval, and they can interact with the system as best suits their particular needs.

Once the user identifies an interesting segment, clicking on its keyframe starts video playback from the beginning of that segment. This ability to start playback at the beginning of a segment is important for exploring a video in depth. It is unlikely that informal video material such as captured meetings would be segmented by hand, as commonly occurs with more formal material such as feature films. Our automatic clustering approach, combined with importance scores, yields segment boundaries that aid video exploration. Further, the interface makes it easy to check a video's promising passages. If a user decides that a passage lacks meaningful content, it's easy to review other segments by clicking on their keyframes.

Many videos in our database depict meetings. If meetings' minutes exist, they can be included in the manga display as captions. We expect that such captions will increase the value of video summaries. Wei Ding and colleagues,<sup>8</sup> for example, reported that participants prefer video summaries that consist of keyframes and coordinated captions and that the combination leads to better predictions of video content. Qian Huang and colleagues<sup>9</sup> have created summaries of news broadcasts in which they use audio and visual



**Figure 3. Web-based, interactive pictorial video summary with highlighted keyframes and embedded captions. The timeline shows the duration of the segment that corresponds with the frame under the cursor.**

characteristics to predetermine story boundaries. For each news story, they extract a keyframe from a portion of the video that contains the most keywords. This method nicely integrates information available for news materials, but it relies heavily on the structured nature of broadcast news and would not apply to general videos.

As Figure 3 shows, we use captions that pop up as the mouse moves over an image. Small icons, such as the letter C, indicate which images have attached captions. The pictorial layout, captioned with text from the meeting's minutes, provides a better summary than the individual parts.

### BROWSING MANY CLIPS

When editing video, extracting good clips from the source material for inclusion in the edited video is a common problem. Clips are usually visualized by one or more keyframes, laid out in a meaningful fashion. When using shots as the smallest unit, finding the desired portion of the video can be difficult because camera motion significantly influences the content over the shot's duration.

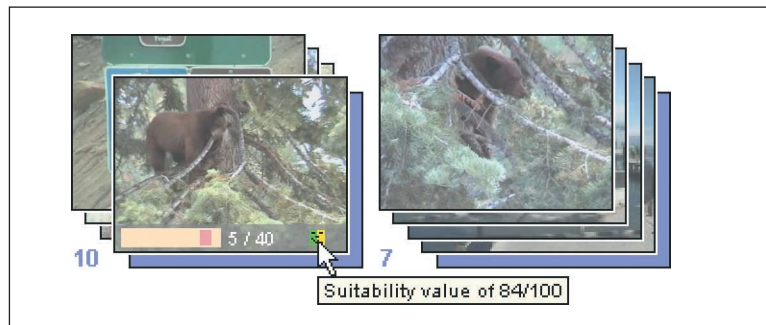
In our system for home video editing,<sup>2</sup> we start with shots determined by a digital-video camera, then subdivide the shots into smaller clips based on camera motion. Mourad Cherfaoui and colleagues<sup>10</sup> produced a system that segments a video into shots and determines if there is camera motion or zooming in each shot. Three frames represent shots that contain camera motion. Their system represents zooms and fixed shots as a single frame with graphical annotations that describe object motion or zoom parameters.

In our system, we break up shots in areas of fast camera motion and present one keyframe for each clip between those areas. Because all of the video's clips must be accessible, we must visualize many clips. We do this by grouping the keyframes associated with clips into piles, based on color similarity. The browser displays the piles row by row in the time order of the first clip in each pile, as Figure 4 shows. Presenting clips in piles rather than as individual images in a scrollable window makes it easier to navigate the organizational structure. The number of piles changes depending on window size.

As Figure 5 shows, the interface moves the image under the cursor to the top of the pile to reveal its content. The user moves the mouse across a pile to flip through a sequence of images representing clips within that pile. For piles containing more than five clips, the browser displays only the first four keyframes together, with a blue frame and a number indicating the pile's height. To navigate through the clip hierarchy, the user clicks on a pile to open an overlay window that displays the contents at the hierarchy's next level.



**Figure 4.** Video editing browser that shows groups of keyframes for a video clip, called piles. The browser sorts the keyframes by color similarity and displays them row by row in the time order of the first clip from each pile. The timeline at the top of the interface shows the coverage of the clips in the pile currently under the mouse cursor, represented by the green area, as well as the start time of the particular clip within that pile, which appears as a red inverted triangle.



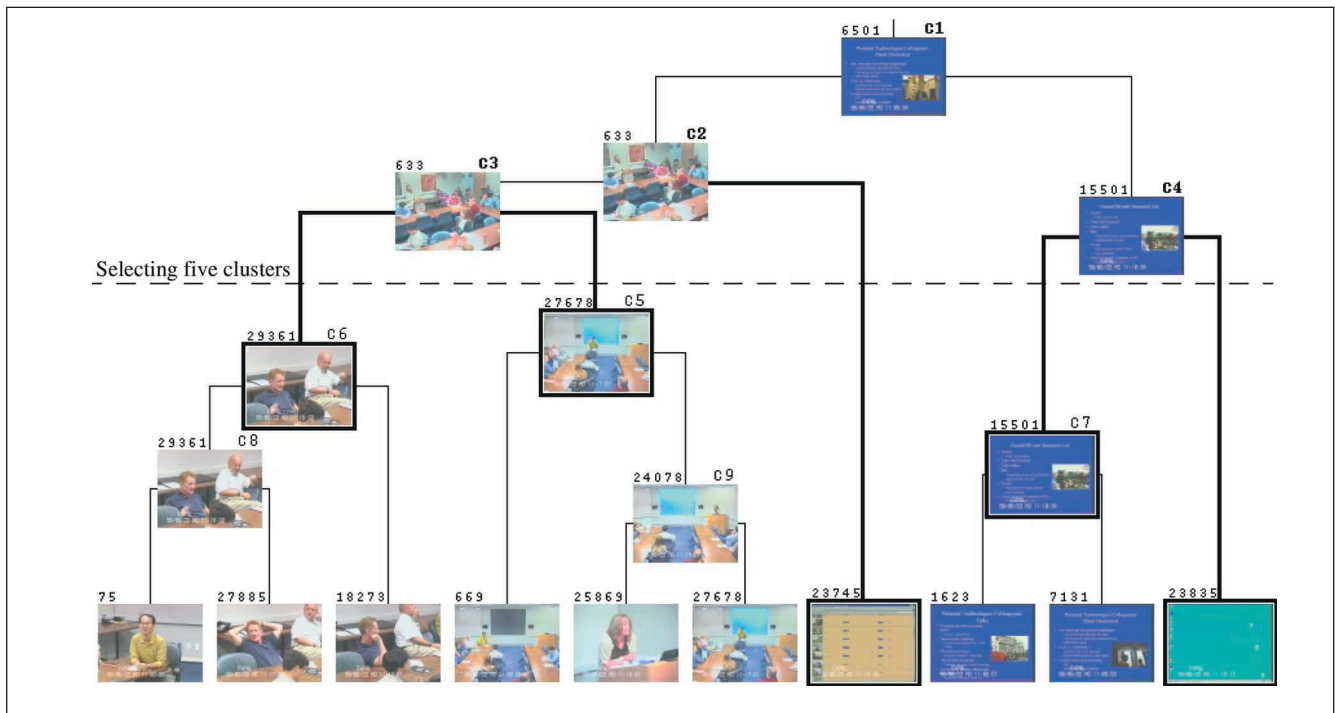
**Figure 5.** When the user places the mouse cursor over an image in the selected pile, the browser moves that image to the top of the pile and provides supplemental information for it with an information bar and tooltip.

Much of the user's effort goes into selecting clips for inclusion in the edited video. One difficulty that users encounter when selecting clips is the lack of information beyond the keyframe and its position in the overall video. To present more information about clips without increasing the required space, we use an information bar and tooltips, shown in Figure 5.

The information bar displays as a translucent overlay for the keyframe and provides the user with the

- length of the clip as a whole,
- length of the default segment of the clip that will be used,
- location of that default segment within the clip, and
- suitability of the default segment.

The left side of the information bar displays a gauge; the size of this gauge is relative to the size of the overall clip. The default segment's position and length appear within the gauge, while text showing the clip's length and the default segment's length appears near the right end of the gauge. A suitability icon, which uses a combination of icon color and shape, appears



**Figure 6. Keyframe tree created using video clustering. A clustering algorithm groups video frames and segments by color similarity, starting with each frame in its own cluster and iteratively combining the two clusters that produce the smallest combined cluster until only a single cluster remains. The member frames closest to the centroid represent clusters.**

near the right end of the bar. Faces, ranging from smiling to frowning expressions with background colors from green to red, indicate five levels of highly to poorly suitable default segments.

In addition to the basic information presented in the information bar overlay, tooltips facilitate the communication of more specific information. A tooltip with general information about the clip, including the take and clip numbers, appears when the mouse pointer lingers over the keyframe. If available, the tooltips include other information provided by the digital-video recorder, such as date and time. When the pointer pauses over the information bar gauge, an explanation of the information in the gauge displays. If the pointer pauses over the suitability icon, a tooltip provides more detailed information about the clip's suitability.

To improve the location of clips, we added a bookmark panel, shown on the right in Figure 4, that provides easy access to clips for use later on. When the user chooses the find clip operation in the composition or bookmark panel, the selection panel displays the hierarchy level where that clip appears by itself and not as part of a pile. In this way, users can navigate to find the context in which they remember other potentially useful clips.

### CLUSTERING

Using the hierarchical agglomerative clustering algorithm<sup>11</sup> to cluster video frames and segments by color similarity is an effective way to select and present keyframes. The algorithm starts with each frame in its own cluster and iteratively combines the two clusters that produce the smallest combined cluster until only a single cluster remains. Figure 6 shows a tree created using this clustering process. The altitude of a node in the tree represents the diameter of the combined clus-

ter—defined as the members' maximum pairwise distance. We use the difference between the color histograms of two frames as the measure of distance.

To select five keyframes in the example in Figure 6, the algorithm splits clusters C1 through C4 and selects one frame from the direct children of the split clusters, as denoted by the images surrounded with bold frames in the figure. Because the video directory listing's slider determines the number of keyframes, this approach sidesteps the problem of selecting an appropriate threshold for the cluster size. When selecting a representative keyframe for a cluster, members of the same cluster should be reasonably similar to one another so that any member can be used. This approach leaves room for applying temporal constraints to the selection of a keyframe from each cluster.

Making the slider interface work well requires a semiuniform distribution in which keyframes are not too close together. The keyframe selection algorithm accomplishes this by iteratively changing the cluster representatives to meet the temporal constraints.

Frame clustering also helps segment a video without resorting to shot detection. The algorithm selects an appropriate number of clusters by plotting the size of the largest cluster against the number of selected clusters and determining the point at which the cluster size changes abruptly. Once the algorithm determines the clusters, it labels each frame with its corresponding cluster. Uninterrupted frame sequences that belong to the same cluster are segments. We use this segmentation approach to create visual video summaries that show one keyframe for each segment, deemphasizing segments that repeat the same scene.

Because our video editing system needs to provide access to every clip, approaches that show a small number of keyframes are inappropriate. We can easily determine one keyframe for each shot because the

camera already performs shot detection. However, a long video can contain a large number of shots, making it difficult to show all the associated keyframes.

Grouping keyframes by color similarity helps present a large number of keyframes. To do this, we use color histogram analysis to cluster the clips hierarchically, visualizing the resulting clusters as piles of keyframes, with a single image denoting each clip. In this situation, we determine the average color histogram for the frames of a shot before clustering the shots. The FotoFile system<sup>12</sup> uses a similar clustering algorithm to present visually similar photographs in a hyperbolic tree, but that approach does not appear to scale up to a large number of images.

**W**e conducted user studies for each of the three user interfaces we've described. In each study, participants preferred our approach for selecting and presenting keyframes over alternatives that did not use our techniques. The studies also provided input for subsequent interface improvements.

The video clip browser is part of a video editor for home users that we are currently improving based on feedback we received in the user study. One challenge is to find a similarity measure for grouping video clips into piles that more closely match human perception. We are also working on providing other grouping mechanisms such as the recording time and on letting users move clips to different piles during the editing process. Another challenge is the further improvement of the video segmentation algorithm based on camera motion that sometimes creates so many clips for long videos that the browsing becomes difficult. The new version will provide users with more information and control without sacrificing the user interface's ease of use. \*

---

## References

1. J. Boreczky et al., "An Interactive Comic Book Presentation for Exploring Video," *Proc. Computer Human Interaction 2000*, ACM Press, New York, 2000, pp. 185-192.
2. A. Girgensohn et al., "A Semi-Automatic Approach to Home Video Editing," *Proc. User Interface Software and Technology 2000*, ACM Press, New York, 2000, pp. 81-89.
3. A. Girgensohn et al., "Facilitating Video Access by Visualizing Automatic Analysis," *Proc. Human-Computer Interaction (INTERACT) 99*, IOS Press, Amsterdam, 1999, pp. 205-212.
4. A.M. Ferman and A.M. Tekalp, "Multiscale Content Extraction and Representation for Video Indexing," *Proc. Multimedia Storage and Archiving Systems II*, SPIE, Bellingham, Wash., 1997, pp. 23-31.
5. Y. Zhuang et al., "Adaptive Key Frame Extraction Using Unsupervised Clustering," *Proc. Int'l Conf. Image Processing 98*, vol. 1, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 866-870.
6. B-L. Yeo and M. Yeung, "Classification, Simplification and Dynamic Visualization of Scene Transition Graphs for Video Browsing," *Proc. Storage and Retrieval for Image and Video Databases VI*, SPIE, Bellingham, Wash., 1998, pp. 60-70.
7. M.G. Christel et al., "Evolving Video Skims into Useful Multimedia Abstractions," *Proc. Computer Human Interaction 98*, ACM Press, New York, 1998, pp. 171-178.
8. W. Ding, G. Marcionini, and D. Soergel, "Multimodal Surrogates for Video Browsing," *Proc. Digital Libraries 99*, ACM Press, New York, 1999, pp. 85-93.
9. Q. Huang, Z. Liu, and A. Rosenberg, "Automated Semantic Structure Reconstruction and Representation Generation for Broadcast News," *Proc. Storage and Retrieval for Image and Video Databases VII*, SPIE, Bellingham, Wash., 1999, pp. 50-62.
10. M. Cherfaoui and C. Bertin, "Two-Stage Strategy for Indexing and Presenting Video," *Proc. Storage and Retrieval for Still Image and Video Databases II*, SPIE, Bellingham, Wash., 1994, pp. 174-184.
11. E. Rasmussen, "Clustering Algorithms," *Information Retrieval: Data Structures and Algorithms*, W.B. Frakes and R. Baeza-Yates, eds., Prentice Hall, Upper Saddle River, N.J., 1992, pp. 419-442.
12. A. Kuchinsky et al., "FotoFile: A Consumer Multimedia Organization and Retrieval System," *Proc. Computer Human Interaction 99*, ACM Press, New York, 1999, pp. 496-503.

*Andreas Girgensohn is a senior research scientist at FX Palo Alto Laboratory. His research interests include user interfaces for video access and editing, Web-based collaborative applications, and video-based awareness systems. He received a PhD in computer science from the University of Colorado at Boulder. Contact him at [andreasg@pal.xerox.com](mailto:andreasg@pal.xerox.com). For more information, see <http://www.fxpal.com/people/andreasg/>.*

*John Boreczky is a senior research scientist at FX Palo Alto Laboratory. His research interests include video content analysis, user interfaces for video browsing, and telepresence. He received an MS in computer science from the University of Michigan. Contact him at [johnb@pal.xerox.com](mailto:johnb@pal.xerox.com).*

*Lynn Wilcox is the manager of the Smart Media Spaces Group at FX Palo Alto Laboratory. Her research interests include audio and video editing, indexing, and retrieval. She received a PhD in mathematical sciences from Rice University. Contact her at [wilcox@pal.xerox.com](mailto:wilcox@pal.xerox.com).*